

Special Instructions for Evidence Copy Box Identification

Documents in this patent application scanned prior to the scan date of this document may not have a box number present in the database. The documents are in the same box as this paper. If the patent application documents that do not have a box number are stored in more than one box, a copy of this form is placed in each box. Check the database box number for each copy of this form to identify all of the evidence copy box numbers for documents that do not have a box number.

☒

The documents stored in this box are original application papers scanned and endorsed by PACT and imported into IFW.

☐

The documents stored in this box were scanned into the IFW prototype for GAU 1634, 2827, or 2834.

Indexer, place and X in only one box above to indicate the documents placed in this box that were previously scanned in PACR or IFW and will not be scanned again.

THIS PAGE BLANK (USPTO)

(12) **UK Patent Application** (19) **GB** (11) **2 349 296** (13) **A**

(43) Date of A Publication 25.10.2000

(21) Application No 9909010.2

(22) Date of Filing 21.04.1999

(71) Applicant(s)
3Com Corporation
(Incorporated in USA - Delaware)
5400 Bayfront Plaza, P O Box 58145, Santa Clara,
California 95052-8145, United States of America

(72) Inventor(s)
Patrick Gibson
Vincent Gavin
Christopher Gilbert

(74) Agent and/or Address for Service
Bowles Horton
Felden House, Dower Mews, High Street,
BERKHAMSTED, Herts, HP4 2BL, United Kingdom

(51) INT CL⁷
H04L 12/56

(52) UK CL (Edition R)
H4K KTK
H4P PPS

(56) Documents Cited
EP 0907300 A **EP 0577359 A2** **WO 99/00949 A**

(58) Field of Search
UK CL (Edition Q) **H4K KTK** , **H4P PPS**
ONLINE DATABASES:WPI, EPODOC, JAPIO

(54) Abstract Title
Reduction of imbalance in transmsit queues in a network switch

(57) A network switch receives addressed data packets and distributes at least some of them into a group of packet queues QA, QB, QC, QD. At least one packet is read out from each queue in turn in a cyclic sequence. A threshold length is defined for each queue such that when a queue length is above its threshold a greater number of packets are transmitted and when the queue length is below the threshold a lesser number of packets are transmitted. More than one threshold 51, 52 may be defined for each queue and the number of packets transmitted from each queue in its turn in the cyclic sequence may progressively increase as the length of the queue exceeds each successive threshold. The address data may be subject to a hashing algorithm. A threshold for each queue may be individually dynamically adjusted according to rate of input traffic, instantaneous queue size or other traffic statistics.

FIGURE 5

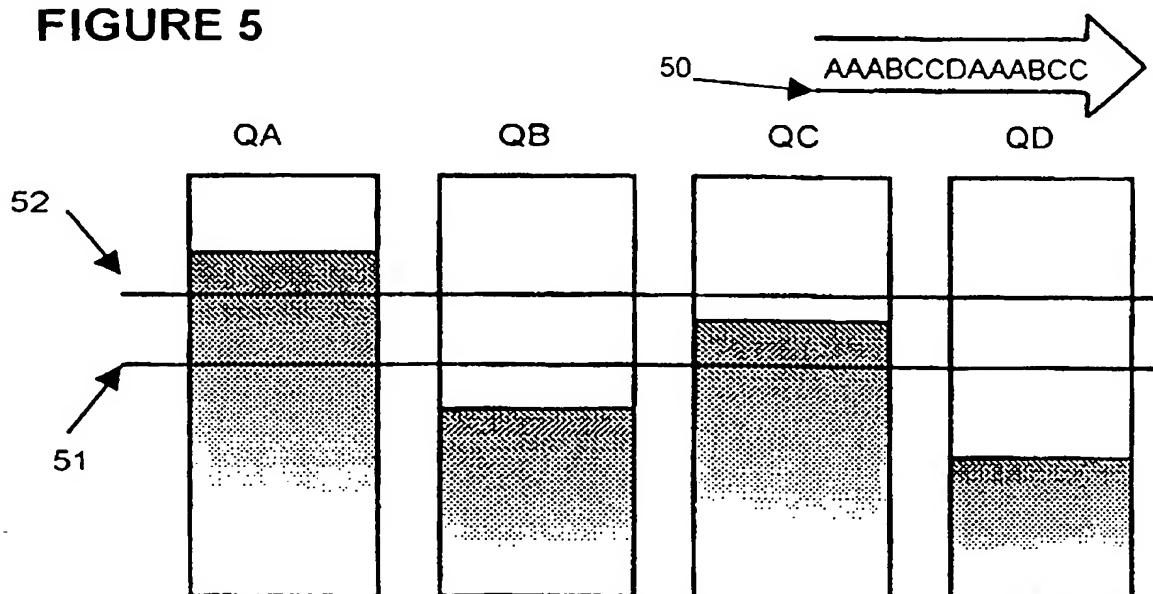


FIGURE 1

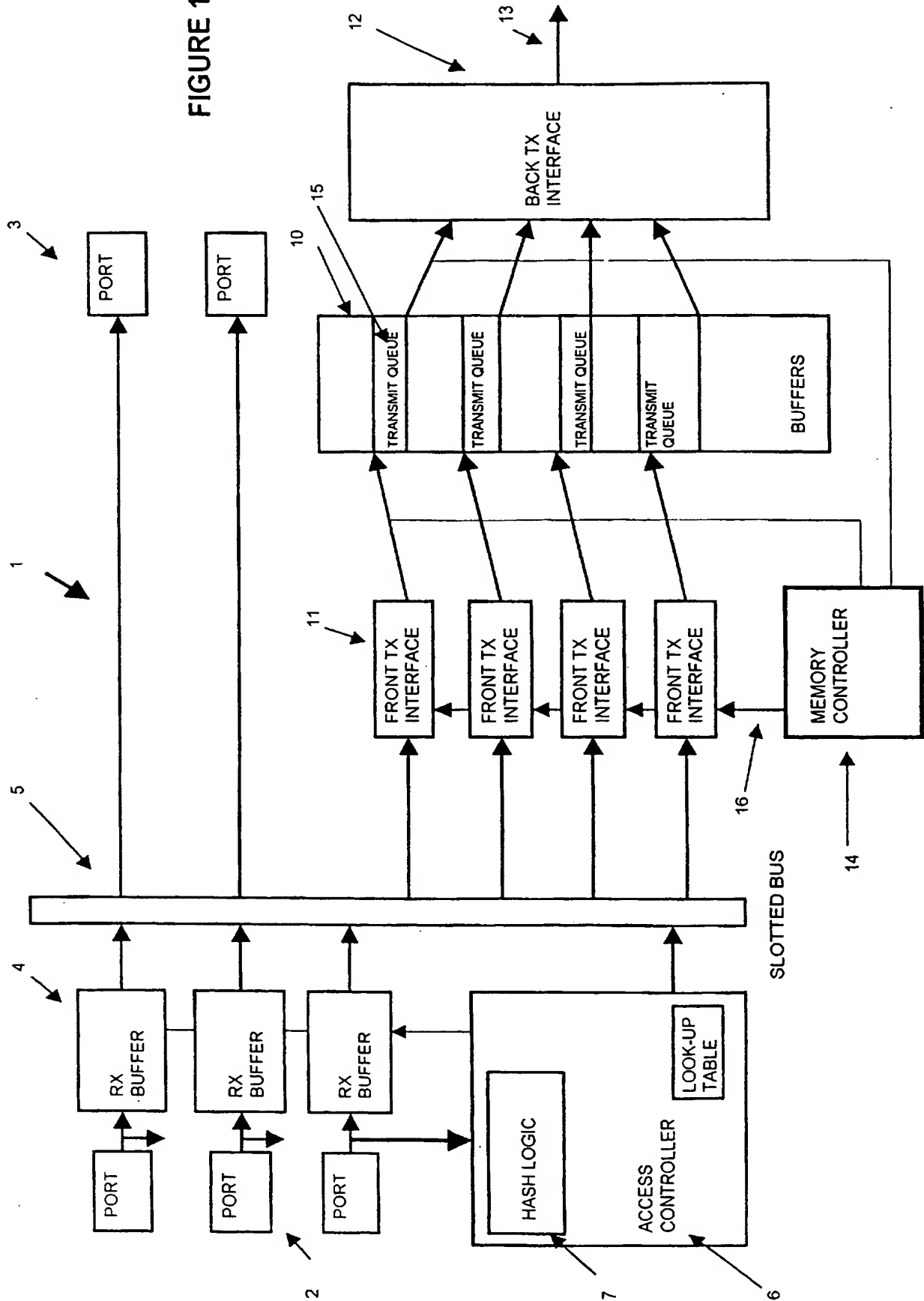


FIGURE 2

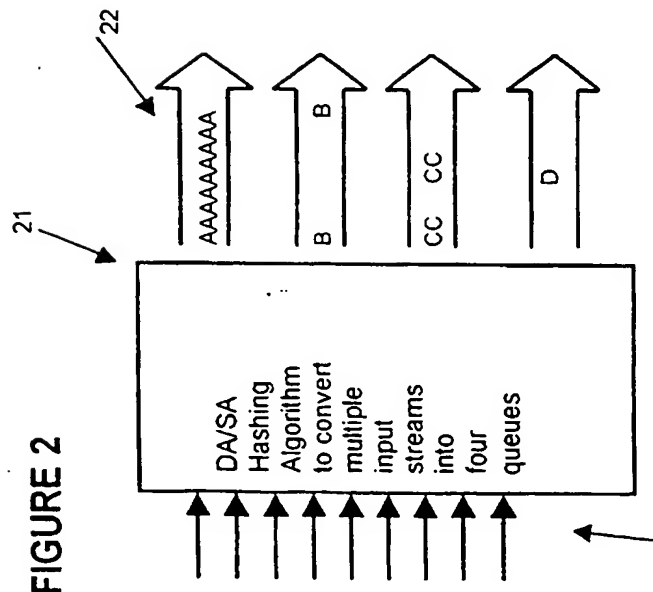


FIGURE 3

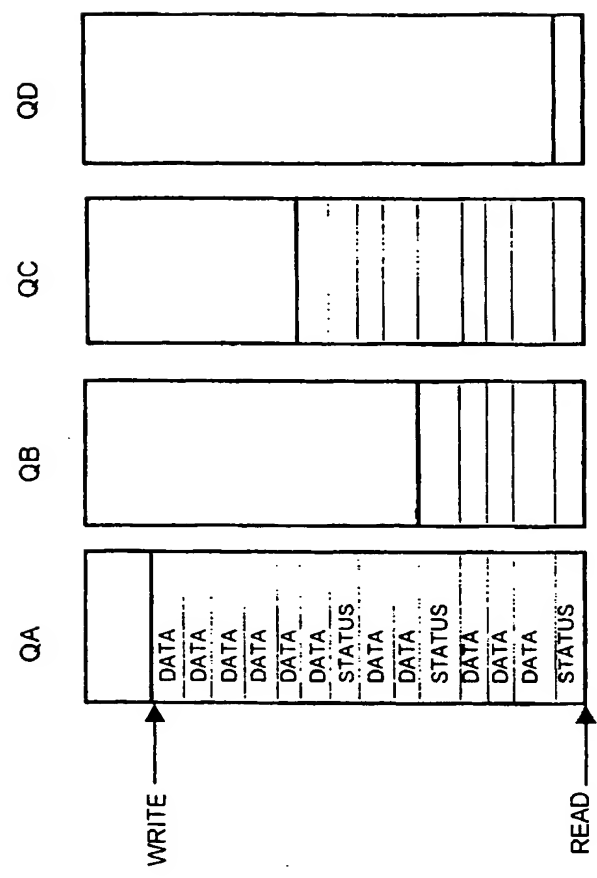


FIGURE 4

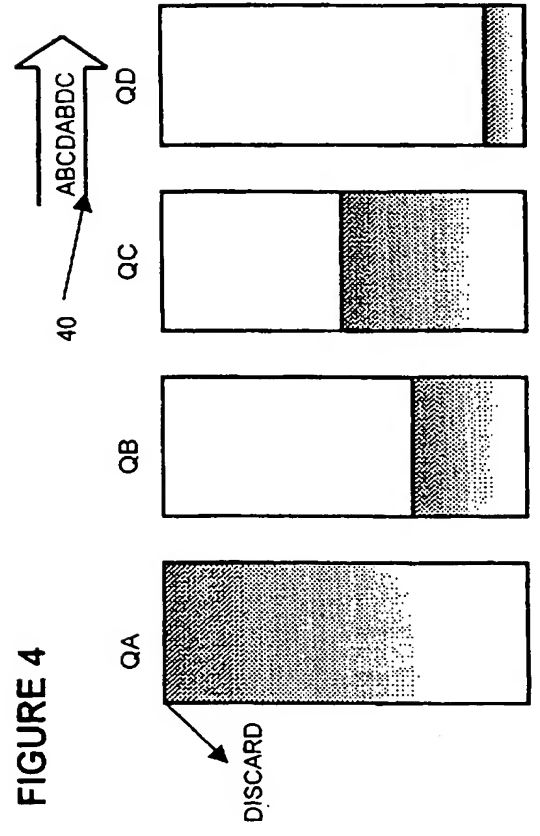
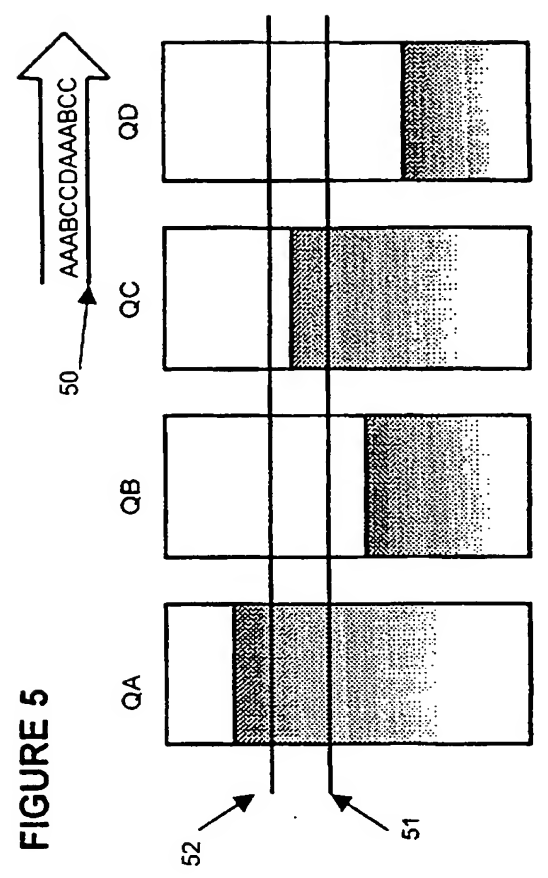


FIGURE 5



REDUCTION OF IMBALANCE IN TRANSMIT TRAFFIC QUEUES IN A NETWORK SWITCH

Field of the Invention

5

The present invention relates to network switches for packet-based communication systems such as Ethernet networks and to an improved method of operating such a network switch. The term 'switch' is intended to refer broadly to a device which receives addressed data packets and which can internally switch those packets in response to that address data or modified forms of such data. The invention is intended to be applicable to a variety of different switch architectures, as indicated hereinafter.

10

Background to the Invention

15

20

25

It is well known to form traffic queues of data packets in network switches. Their formation is necessary to provide temporal buffering of a packet between the time it is received at a network switch and the time at which it can be transmitted from the switch. In most forms of network switch, the switch has a multiplicity of ports, which are normally capable of duplex working, and data packets received at the ports may after appropriate processing including look-ups in relation to destination and source addresses in the packets, be directed to a port or ports in accordance with that address data. Switches employing both media access control addresses (such as in bridges) or network addresses (such as in routers) are of course well known in the art. In such switches it is customary to provide temporal buffering both when the packets are received, in what are known as 'receive queues' and when they are assigned to transmit ports, known as 'transmit queues'. In general, the transmission of packets from a transmit queue may depend on a variety of considerations, including possible congestion in a device to which the respective port is connected.

30

A particular example of a multiplicity of traffic queues, relevant to the present invention, is the formation of a group of traffic queues, typically four or more, of packets received at a port or ports of a switch. packets being read out from the queues by means of an interface

which performs a read-out from each of the queues in turn. One reason for so doing is to provide the functionality of a group of 'trunked' ports from which data packets, read out in cyclic sequence, may be transmitted from the switch along a single high speed link.

5 It is known to form queues of data packets in a variety of ways, including comparatively simple FIFOs established in hardware. More usually in modern switches queues may be formed in random access memory employing read and write pointers under the control of a memory controller. If static random access memory is employed, a particular traffic queue may be allotted a defined memory space and packets may be read in to that memory space
10 under the control of a read pointer which progresses from one location to another until it reaches the 'end' of the allotted memory space whereupon it recycles to the beginning of the memory space (on the assumption that the space is not fully occupied). A read pointer progresses through the memory space in a similar manner. In such systems the fullness of a memory space or thresholds representing some fraction of fullness need to be expressed in
15 terms of the effective distance in terms of memory locations between the read and write pointers.

Another and preferred system in a specific example of the present invention is a dynamic memory comprising a plurality of identifiable buffers which can be allotted to a specific
20 traffic queue under the control of a Free Pool Controller and Transmit (Tx) Pointer Manager, termed for convenience herein 'memory controller'. In such a system, any particular traffic queue may have initially some small number, such as two, buffers allotted to it. If a queue requires more traffic space, then the memory controller can allot additional buffers to the queue. It is, as indicated for the previous example, possible to limit the
25 available memory space by a limitation on the number of buffers employed for any particular queue, though it is known, and preferable in a variety of circumstances, to allow some traffic queues more space than others by imposing a different limit on the maximum number of buffers which can be used for that queue. In buffer systems, data may be written into the buffers using a write pointer and read out from the relevant buffers using a read pointer. In
30 general, the size of each buffer is substantially more than that of a single packet. Packets are normally stored in such buffers in the form of a status word (which would normally be read

first), including some control data and also an indication of the size of the packet, followed by address data and message data. An interface which reads a packet from such a buffer store will, in a reading cycle, commence reading the status word and proceed to read the packet until the next status word is reached.

5

One example of a dynamic buffer memory which has thresholds for limiting the respective number of buffers available for each of a plurality of traffic queues is disclosed in British patent number 2321820 granted to 3Com Technologies.

10 It is also known to distribute packets in a switch by means of hashing of address data in received packets. Hashing is known to facilitate the processes of look-up (to determine the port or traffic queue to which a packet should be sent). A variety of hashing algorithms, generally involving combinatorial operations such as exclusive-or on a selection of bits of address data are known. Since packets received at a particular network switch may be
15 dominated by a particular source (such as a file server or perhaps a single user), it is commonplace to employ hashing on a combination of both the source address and the destination address in a packet.

In the particular circumstances described above, wherein a plurality of traffic queues are
20 intended to provide packets for ultimate transmission over a single common link, hashing provides a means of distributing packets to the queues and ideally the distribution would be perfect so that queues were in practical terms of substantially the same length. In the ideal case, that facilitates the most efficient use of storage capacity. Nevertheless, in any practical system the length of individual traffic queues in a set of traffic queues will vary considerably
25 partly due to hashing, and possibly due to other considerations such as certain types of traffic being given priority in a contentious process over packets of lower priority. Although dynamic buffering may be employed to accommodate different lengths of traffic queues of packets ready for transmission, there are practical and economic limits on the use of dynamic buffering. Even when dynamic buffering is employed and different lengths of traffic
30 queue can be accommodated, there must in any practical system be a limit to the number of buffers, or in general the amount of memory space, allotted to a traffic queue. When this

limit is reached, the packets intended for that traffic queue must be discarded. There are several different mechanisms for achieving discard of packets. Discard is normally initiated when the occupancy of the maximum allotted space to a traffic queue approaches the physical limit. It may in some circumstances be desirable to set the limit at which discard of packets is initiated at less than the absolute physical limit but this variation is not of importance to the invention. Discard of packets may be effected either by inhibition of the writing process by an interface which would normally write packets into the respective transmit queue under the action of a free pool controller but may alternatively be achieved by signalling across the switch to effect discard of packets intended for the respective transmit queue as or before they are sent from the receive queues.

Summary of the Invention

The problem to which the present invention is directed is the reduction of unnecessary discarding of packets when a particular transmit queue reaches the limit of the respective available memory capacity (whether dynamically adjusted or not) in circumstances where there is an imbalance of lengths of the transmit traffic queues in a group and there is therefore memory space still available in respect of the other traffic queues which are of lesser length or at least which occupy a lesser proportion of their allotted memory space. The problem typically arises by virtue of the use of hashing address data but the invention is not necessarily limited to that circumstance.

One aspect of the present invention comprises defining for each of the traffic queues in a group at least one threshold representing a specified proportion of fullness of the respective memory space and, when such a threshold is exceeded, enabling weighted read-out of that particular traffic queue. In particular, an interface which executes the read-out may be enabled, if the respective threshold is exceeded, to read out two or more packets in that queue's turn instead of only one packet per turn. Although such a differential rate of transmission of packets from the queues will not necessarily prevent the discarding of packets, the rate of discard from heavily loaded and therefore longer queues may be reduced and there will be a greater tendency for the traffic queues to occupy the same

proportion of their respective allotted memory space or allotted maximum number of buffers however that memory space is defined.

5 It is possible to define for each traffic queue more than one threshold, each threshold defining a greater proportion of fullness of the respective maximum memory space and to increase further the number of packets transmitted in each turn as each successive threshold is exceeded.

10 Another important aspect of the invention is to utilize the foregoing functionality, and to monitor various characteristics or parameters of the traffic queues, such as not only the instantaneous size of the queue but also the rate of input traffic and to adjust the threshold or thresholds defined for each queue in accordance with the monitored statistics. In particular, a threshold could be lowered, thus advancing the onset of multiple transmission of packets from that queue in its turn, if the monitoring of the statistics of the traffic
15 indicated that the rate of packets directed to that queue was increasing. Various other dynamic adjustments of the thresholds would be feasible. For example, the thresholds may be adjusted by means of a control algorithm which includes such variables as the instantaneous size of a queue and the data rate (e.g. bytes/second).

20 These and other features of the invention will be explained with reference to a specific example and with reference to the accompanying drawings.

Brief Description of the Drawings

25 Figure 1 illustrates in general terms the architecture of a network switch within which the present invention can be practised;

Figure 2 illustrates the results of an unbalanced hash of addresses;

30 Figure 3 illustrates by way of example an initial state of four traffic queues;

Figure 4 illustrates the effects of a regular round robin read-out from the traffic queues shown in Figure 3; and

Figure 5 illustrates an improved method of read-out from traffic queues in accordance with the invention.

Detailed Description

Figure 1 shows, by way of example only, a typical architecture in which the present invention may be practised.

The drawing illustrates a network switch 1 which has a set of ports 2 receiving addressed packets such as Ethernet packets. Very typically there are many more ports than three and also typically the ports are capable of duplex working, namely they can transmit as well as receive. However, for the sake of simplicity, the ports 2 are shown as receiving ports, further ports 3 are shown as transmitting ports and the switch includes provision for a high speed link, preferably constituted by a serial link 13, which can transmit data selected in turn from a multiplicity of traffic queues (four in this example) established as hereinafter explained in a buffer memory 10.

Packets received at each of the ports 2 are temporarily stored in receive (RX) buffers 4 from which they may be distributed to their destination ports by any convenient means. In this example, the receive buffers provide packets under the control of an access controller 6 to a time slotted data bus 5 which provides equal access time for each of the receive buffers. From the data bus packets are delivered either to the 'transmitting' ports 3 or to one of four interfaces 11 which constitute output ports as far as the slotted bus is concerned. All the explicit ports 3 and the interfaces are allotted 'port numbers'. The distribution of port numbers to packets is achieved in known manner by applying a hashing algorithm (shown within the access controller 6) to at least part of the address data in the received packets and preferably on a combination of the source address and destination address in each packet.

This may readily be achieved in known manner by hashing the source and destination addresses to obtain a pointer to an entry or series of linked entries, in a look-up table 8 which typically contains entries each comprising at least a 'port number' and usually the original source and destination address data for the purpose of verification and also a link pointer to other entries in the table obtained from different source and destination pairs but hashing to the same result, so that if a verification process made on an entry does not achieve a match of address data the next entry in a link list can be examined and so on. This process is typical of look-ups for forwarding tables based on hashing of addresses and need not be described in detail. One example of hashing is described in US patent No. 5708659 to Rostoker et al, issued 13 January 1998.

In any event, those packets which are allotted port numbers corresponding to any of the ports 3 will after leaving the relevant RX buffer be conveyed across the switch by way of the slotted bus to those ports 3.

In place of four ports having ordinary traffic queues, there are four front Tx interfaces 11 which are allotted port numbers so that packets which the hashing process allots the relevant port numbers will be coupled by way of the slotted bus to these interfaces, the purpose of which is only to effect the writing in of the packets to a respective one of a plurality of transmit queues, one for each of the 'front TX interfaces'.

Each transmit queue (such as queue 15) is disposed in one or more buffers in a large buffer store 10 (constituted for example by dynamic random access memory) under the action of a memory controller 14 which, in known manner, controls the generation and positioning of write pointers and read pointers. As has already been generally described, each front Tx interface writes each new packet into a respective buffer as directed by the respective write pointer. The memory controller 14 may initially allot some selected minimum of buffer space to each queue formed by the packets successively received by each respective interface. Typically, the free pool controller will initially allot two buffers to each traffic queue. Normally the buffer size is equal to several maximum size data packets, which typically can vary in size from about 64 bytes up to 1.5 kilobytes.

Whatever scheme is adopted for holding the transmit queues, whether the buffer system described, or the provision of defined memory space in static random access memory, or otherwise, the controller such as the memory controller 14 normally includes means for detecting the relative fullness of each memory space. This may be defined as less than the maximum allotted space but in practical terms the space is defined by a predetermined limit (which may be dynamically adjustable) such that if the limit is reached by the data packets in the queue, the controller provides a signal which will initiate the discard of packets intended for that queue. The discarding function may be performed in a variety of known ways, one of which is simply to prevent the respective front TX interface 11 from accepting any further packets while the respective transmit queue remains full. This is illustrated schematically in Figure 1 by the control line 16 extending from the memory controller 14. There are however other ways of performing the discard function. One of them is to signal across the slotted bus to the access controller 6 to prevent the transmission to the slotted bus of packets which will otherwise be placed on the bus 5 for transmission to the respective front TX interface 11.

Packets are read out from the queues under the control of read pointers likewise under the action of the memory controller. The pointers point (again in a manner known in itself) to the 'oldest' packet in any particular transmit queue and define the start point of a reading cycle for a back TX interface 12. This interface reads out from the transmit queues in a round robin manner, that is to say from each of the transmit queues in turn. As previously remarked, the read pointers indicate the commencement of a reading process by the back transmit interface. As each buffer is cleared of data by the read process, it becomes a 'free' buffer which can be allotted by the memory controller to a transmit queue when needed. In buffer control systems of this nature, there is not a dedication of particular buffers to particular queues; in principle any buffer may be allotted to any queue and the limitation of the size of a queue is determined by the number and not the location of the buffers. In other memory systems such as the one described above using dedicated space in SRAM, a specific memory space is allocated to a traffic queue and the reading and writing pointers recirculate around the dedicated space

The reading of data from the traffic queues in round robin manner is a practical necessity in systems wherein packets from a multiplicity of queues need to be 'trunked', namely sent down a common high speed link, such as a serial data link 13.

Before leaving Figure 1, the skilled reader will appreciate that it is common practice to employ software control for the buffer memory by the memory controller. The memory controller essentially maintains, in this example, four lists of buffers allotted to the queues and identifying words defining the pointers and their positions. The pointers are employed by the interfaces as a means of addressing the memory to write information in and read information out of the buffer memory at the places indicated.

Figure 2 is a functional summary of the operation of the architecture shown in Figure 1. A multiplicity of inputs 20 are subject to a DA/SA hashing algorithm 21 to convert the multiple input streams of packets, received at ports of the switch, into, for present purposes, four output streams 22 containing packets A, for a queue QA in Figure 3, packets B for a queue QB in Figure 3, packets C for a third queue QC in Figure 3, and packets D for a fourth queue, QD in Figure 3.

Figure 3 illustrates a typical instantaneous view of the group of packet queues QA-QD. The disposition of packets in the queues is similar and is illustrated in detail only for the first queue QA. Each packet consists of a status word followed by address data and the remains of the packet. The next packet is delimited by its status word and so on. It will also be understood that the buffer queues shown in Figure 3 are actually dynamic. The write pointer is proceeding through the buffer memory as buffers are allotted to that queue. As the read pointer proceeds buffers are freed in their turn and returned to the 'free pool' of buffers available for allotment to one or other of the queues of packets.

As will be apparent from Figure 3, the first queue, which is receiving the packets at the greatest rate, is likely to reach the limit of its allotted space more rapidly than the others. When it does so, the memory controller 14 will initiate, for example by a control to the relevant interface 11, the discard of packets.

Figure 4 illustrates therefore what would happen if packets were read out in a simple round robin fashion, as shown by the accompanying arrow, illustrating the output of packets from each of the queues one at a time in order, namely ABCD ABCD. The packets in the first queue QA will be discarded even though there is substantial space available in the memory for the other queues QB, QC and QD.

A partial solution to the problem is to allow the first queue to grow in size in accordance with the traffic demands for it. This is not a complete solution, for several reasons. First, there has to be a practical maximum size on a buffer queue. Second, merely allowing the queue to become disproportionately large does not cure the basic trouble, which is that packets are entering the queue faster than they can be removed from it.

The solution according to the present invention comprises defining for each of the queues at least one threshold which can be used to alter the read-out of the queues by the read-out interface. By way of example, Figure 5 illustrates two thresholds 51 and 52, which are shown as common to all the queues, defined for different fractions of fullness of the queues. If a queue is below the first threshold, read-out from that queue will proceed normally, one packet in that queue's turn. If a queue is above the first threshold, there will be a read-out of two packets per turn for that queue. If the queue is beyond the second threshold, there will be a read-out of three packets per turn, as illustrated by the progression of packets in arrow 50. In this way, as each queue becomes progressively fuller, the rate of read-out of packets increases.

This scheme has several benefits. First, the weighted read-out produces a greater tendency to equality in the space occupied by the buffers and therefore utilises the buffer space more efficiently. Second, queues which receive packets at a higher rate are automatically allotted a greater proportion of message space in the output link. A further benefit of a scheme as described herein is that it is adaptable to further refinement by monitoring of the fluctuating statistics of the queues, that is to say an input rate rather than merely the instantaneous size of the queue. In particular, monitoring of the packets can readily be achieved at for example the input of the front interfaces. If a rate of input or in the alternative an average packet size

increases, the thresholds for that particular queue may be lowered so that the onset of multiple packet transmission at each turn of that packet queue occurs for a lesser fraction of fullness of the queue.

5

10

15

20

25

30

CLAIMS

1. A method of operating a network switch which has means for receiving addressed data packets and distributes at least some of the received packets into a group of packet queues, the method comprising:

(a) reading out at least one packet from each queue in the group in a cyclic sequence;

(b) defining for each queue at least one threshold representing a selected length for the queue;

(c) transmitting from a queue in each turn a lesser number of packets while the length of the respective queue is less than the threshold; and

(d) transmitting a greater number of packets from the respective queue in each turn when the length of the queue exceeds the threshold.

2. A method according to claim 1 wherein more than one threshold is defined for each queue and wherein the number of packets transmitted from each queue in its turn in the cyclic sequence progressively increases as the length of the queue exceeds each successive threshold.

3. A method according to claim 1 wherein address data in the packets is subject to a hashing algorithm to direct the distribution of the packets to the said queues.

4. A method according to claim 1 wherein a threshold for each queue is individually adjusted.

5. A method according to any foregoing claim wherein the packets from the queues are transmitted successively in a common output.

6. A network switch comprising:

means for receiving addressed data packets and for distributing at least some of the received packets into a group of packet queues;

5

means for reading out at least one packet from each queue in the group in a cyclic sequence; and

means for defining for each queue at least one threshold representing a selected length for the queue;

10

wherein said means for reading out is operative to read out from each queue in each turn a lesser number of packets while the length of the respective queue is less than the threshold; and is operative to read out a greater number of packets from said respective queue in each turn when the length of the respective queue exceeds the threshold.

15

7. A switch according to claim 6 wherein more than one threshold is defined for each queue and wherein the number of packets transmitted from each queue in its turn in the cyclic sequence progressively increases as the length of the queue exceeds each successive threshold.

20

8. A switch according to claim 6 wherein and including means for transmitting the packets from said queues in a common output.

25

30